Hepatoprotective effects of systemic ER activation

Single cell analysis

Christian Sommerauer & Carlos Gallardo

25 July, 2023

```r
# source and library import
source('code/00_helper_functions.R')
library(tidyverse)
library(Seurat)
library(pagoda2)
library(patchwork)
```

```r
# color palettes
colPals <- list()
colPals$conditions <- setNames(c('#44AA99', '#117733', '#88CCEE', '#332288', '#DDCC77', '#CC6677', '#AA4499', '#882255'),
                               c('CDf', 'HFDf', 'CDm', 'HFDm', 'DPN', 'DIP', 'E2', 'PPT'))
colPals$RdBu <- rev(RColorBrewer::brewer.pal(n=11, name = 'RdBu'))
colPals$UpDown <- setNames(colPals$RdBu[c(10,2)],
                           c('up', 'down'))
colPals$clusters <- setNames(c('#E6E6E6', '#B3B3B3', '#8C8C8C', '#4D4D4D'),
                             c('1', '2', '3', '4'))
colPals$celltypes <- setNames(c('#B4272F', '#E5462D', '#FFD1D1', '#F4E54C', '#FBAA3E', '#AA654E', '#B58B80', '#6D7AA5',
                                '#B3177E', '#CAC1DD', '#67227D','#36B449', '#82C349', '#A9D265', '#199478', '#95A3A3', '#C4C5C7'),
                              c('Cholangiocytes', 'Endothelial cells', 'HsPCs', 'Stromal cells', 'Hepatocytes',
                                'Kupffer cells', 'Monocytes & Monocyte-derived cells', 'T cells', 'NK cells',
                                'ILC1s', 'B cells', 'cDC1s', 'cDC2s', 'Mig. cDCs', 'pDCs', 'Basophils', 'Neutrophils'))
colPals$inferno <- c('#FCFFA4', '#FCA50A', '#DD513A', '#932667', '#420A68', '#000004')
```

Note

This script is highly demanding in terms of computational resources. Approximately 50GB of single-cell data from the Liver Cell Atlas resource are loaded into memory. Establishing cell enrichments of gene signatures with `pagoda2` also requires long computing time. Run using `pagoda2` v1.0.2 and `Seurat` v4.0.2 on R version 4.0.5

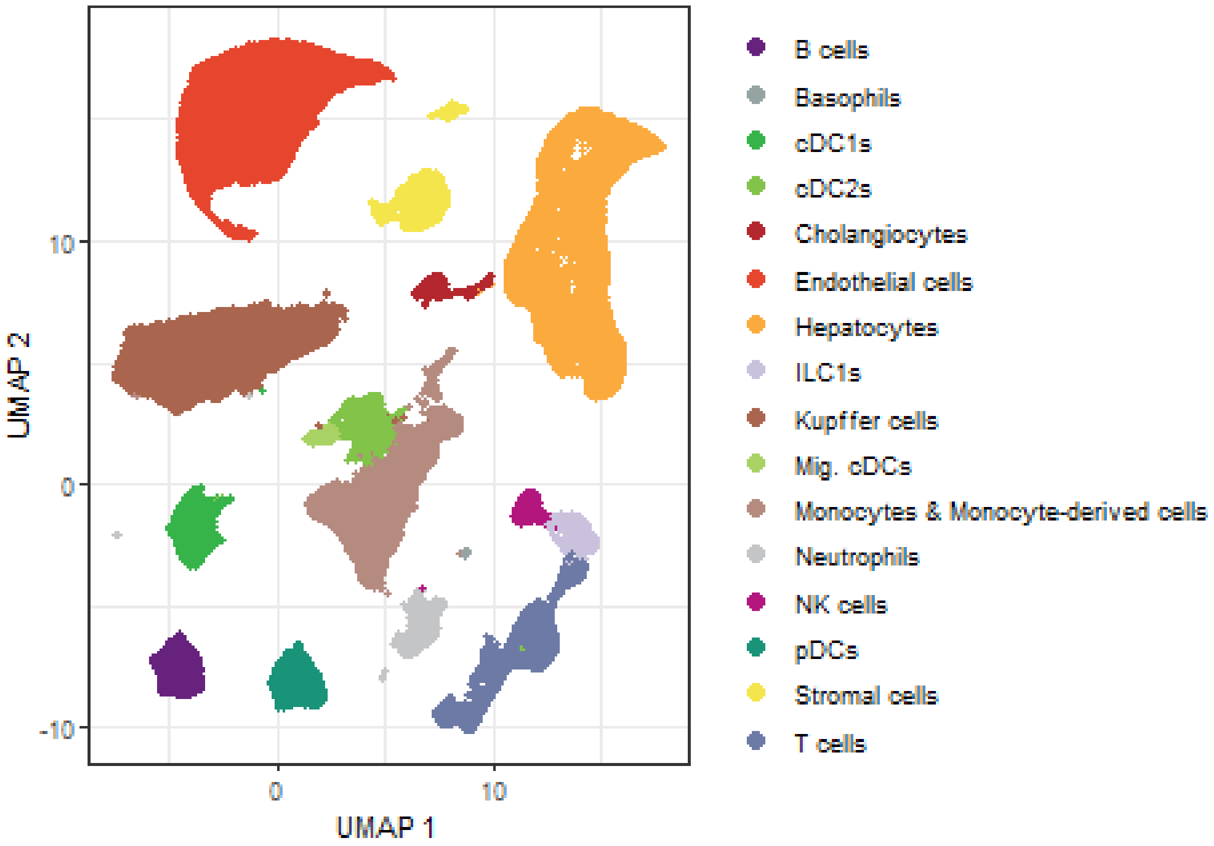
Load data

```r
# Liver Cell Atlas data from Guilliams et al. 2022 (https://www.livercellatlas.org/download.php)
liver <- readRDS(file = 'data/livercellatlas_feb2022.rds')

# Liver Cell Atlas sample metadata
liver_meta <- read.table(
  file = 'data/livercellatlas_feb2022_sample_annotation.tsv',
  stringsAsFactors = FALSE,
  sep = '\t',
  header = TRUE)

# gene sets
DEG_sets <- readRDS('results/bulkRNAseq_mmus_DEG_sets.rds')
pathway_sets <- readRDS('results/bulkRNAseq_mmus_GSEA_reactome_cluster_sets.rds')

# mouse-human orthologs
mouse_human_orthologs <- read.table(
  file = 'data/ensembl_mmus_hsap_sep2019_orthologs.tsv',
  sep = '\t',
  header = TRUE,
  quote = '')

# mouse-macaque orthologs
```

```
mouse_macaque_orthologs <- read.table(
  file = 'data/ensembl_mmus_mfas_sep2019_orthologs.tsv',
  sep = '\t',
  header = TRUE,
  quote = '')
```

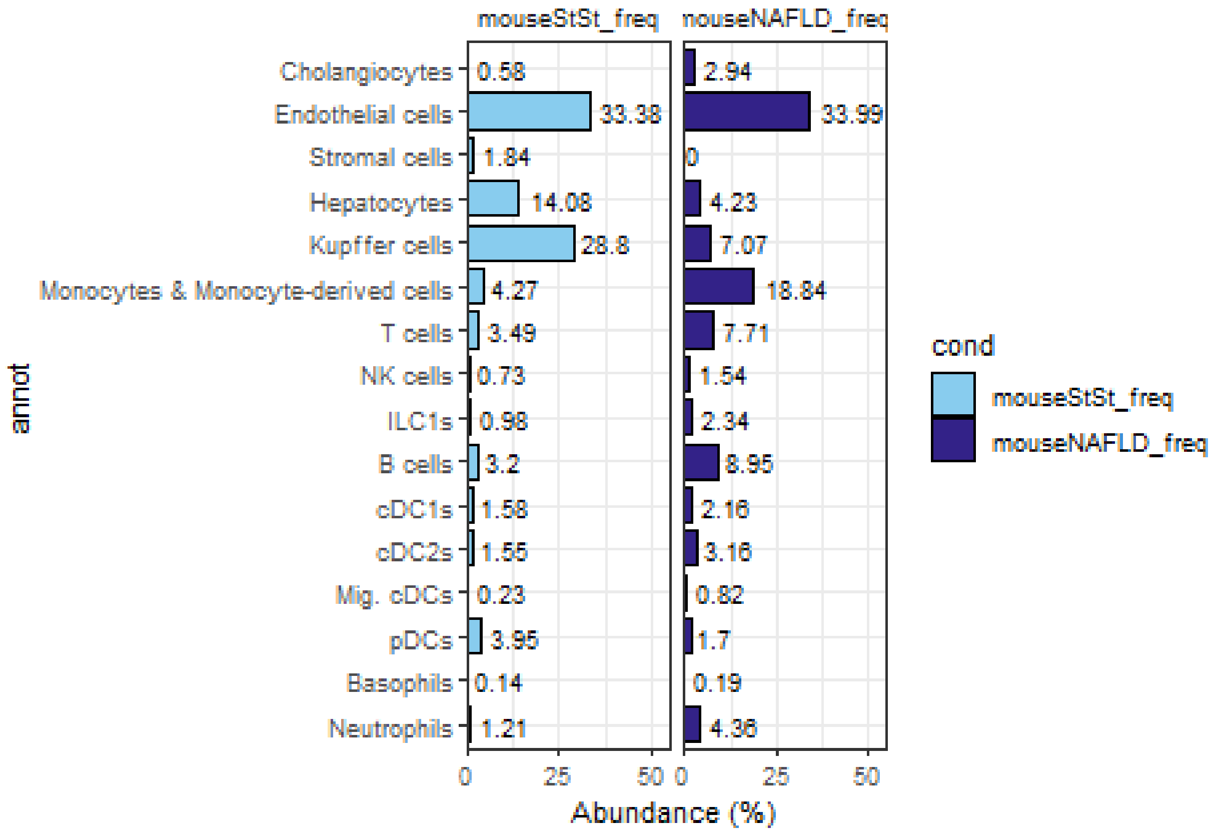
Mouse single-cell map

```
# filter for male-only samples and remove HsPCs
df <- liver$mouseStSt$All@meta.data %>%
  dplyr::left_join(liver_meta, by = 'sample') %>%
  dplyr::filter(sex == 'male' & annot != 'HsPCs') %>%
  dplyr::mutate(annot = replace(annot, annot == 'Fibroblasts', 'Stromal cells'))

ggplot(df, aes(x=UMAP_1, y=UMAP_2, color=annot, fill=annot, label=annot)) +
  geom_point(shape=21, size=1) +
  scale_fill_manual(values = colPals$celltypes) +
  scale_color_manual(values = colPals$celltypes) +
  xlab("UMAP 1") + ylab("UMAP 2") +
  theme_bw() +
  guides(colour = guide_legend(override.aes = list(size=3))) +
  theme(legend.title = element_blank())
```



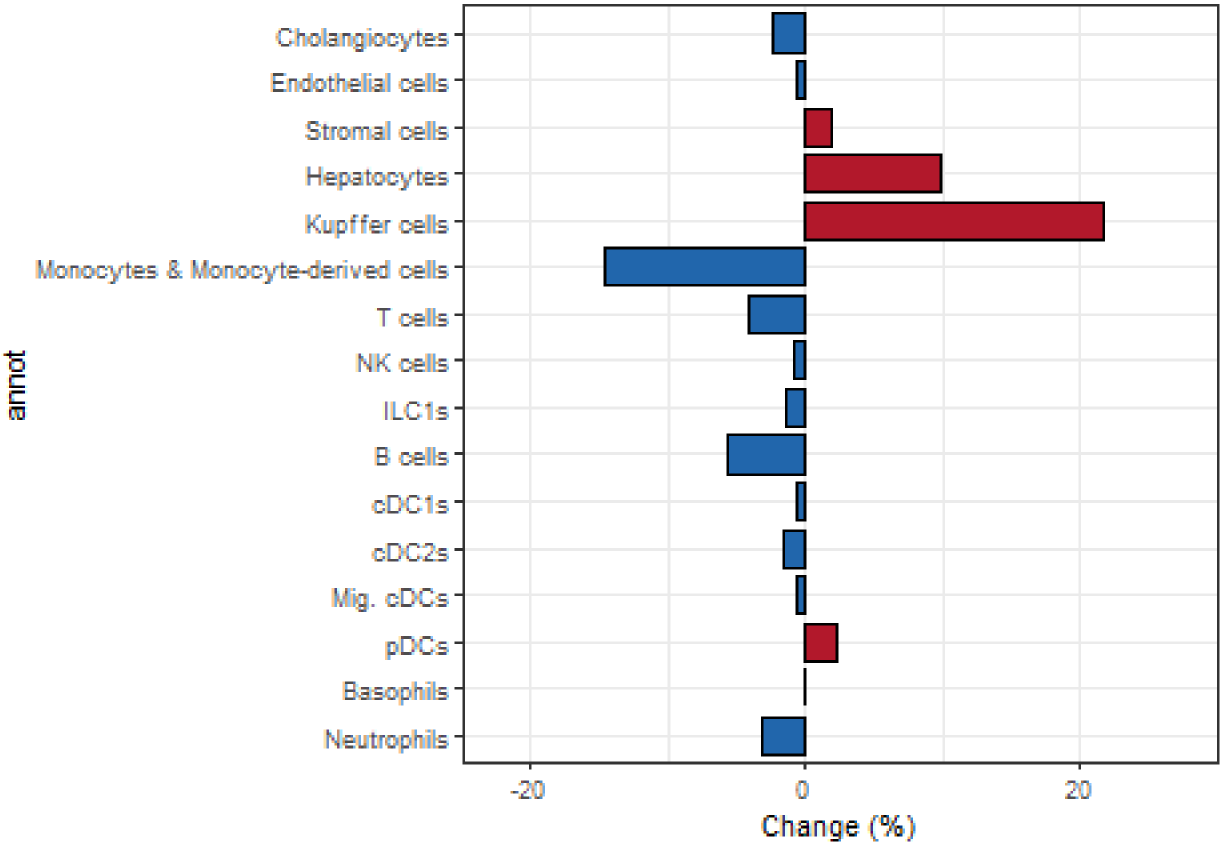
Cell type composition

```
# filter for male-only samples and remove HsPCs
# filter for cells from in vivo digestion protocol
df <- liver$mouseStSt$All@meta.data %>%
  dplyr::left_join(liver_meta, by = 'sample') %>%
  dplyr::filter(sex == 'male' & annot != 'HsPCs') %>%
  dplyr::mutate(annot = replace(annot, annot == 'Fibroblasts', 'Stromal cells')) %>%
  dplyr::filter(digest == 'inVivo') %>%
  dplyr::group_by(annot) %>%
  dplyr::summarize(mouseStSt_n = dplyr::n()) %>%
  dplyr::mutate(mouseStSt_freq = round(mouseStSt_n / sum(mouseStSt_n) * 100, 2))

df2 <- liver$mouseNAFLD$All@meta.data %>%
  dplyr::left_join(liver_meta, by = 'sample') %>%
  dplyr::filter(sex == 'male' & annot != 'HsPCs') %>%
  dplyr::filter(digest == 'inVivo') %>%
  dplyr::mutate(annot = replace(annot, annot == 'KCs', 'Kupffer cells')) %>%
  dplyr::mutate(annot = replace(annot, annot == 'Mono/mono-derived cells', 'Monocytes & Monocyte-derived cells')) %>%
  dplyr::mutate(annot = replace(annot, annot == 'Mig cDCs', 'Mig. cDCs')) %>%
  dplyr::group_by(annot) %>%
  dplyr::summarize(mouseNAFLD_n = dplyr::n()) %>%
  dplyr::mutate(mouseNAFLD_freq = round(mouseNAFLD_n / sum(mouseNAFLD_n) * 100, 2))

df <- dplyr::full_join(df, df2, by = 'annot') %>%
  replace(is.na(.), 0) %>%
  dplyr::mutate(mouseStSt_minus_NAFLD_freq = mouseStSt_freq - mouseNAFLD_freq) %>%
  tidyr::pivot_longer(cols = c(mouseStSt_freq, mouseNAFLD_freq), names_to = 'cond', values_to = 'freq') %>%
  dplyr::mutate(annot = factor(annot, levels = rev(names(colPals$celltypes))),
              cond = factor(cond, levels = c('mouseStSt_freq','mouseNAFLD_freq')))

ggplot(df, aes(x=freq, y=annot, fill=cond, label=freq)) +
  geom_bar(color="black", size=0.5, width=0.8, position="stack", stat="identity") +
  geom_text(size=3, hjust=-0.2) +
  scale_fill_manual(values = unname(colPals$conditions[c('CDm', 'HFDm')])) +
  scale_x_continuous(expand = expansion(mult = c(.0, .1)),
                     limit = c(0, 50),
                     breaks = c(0, 25, 50)) +
  facet_wrap(~cond) +
  xlab('Abundance (%)') +
  theme_bw() +
  theme(strip.background = element_blank())
```



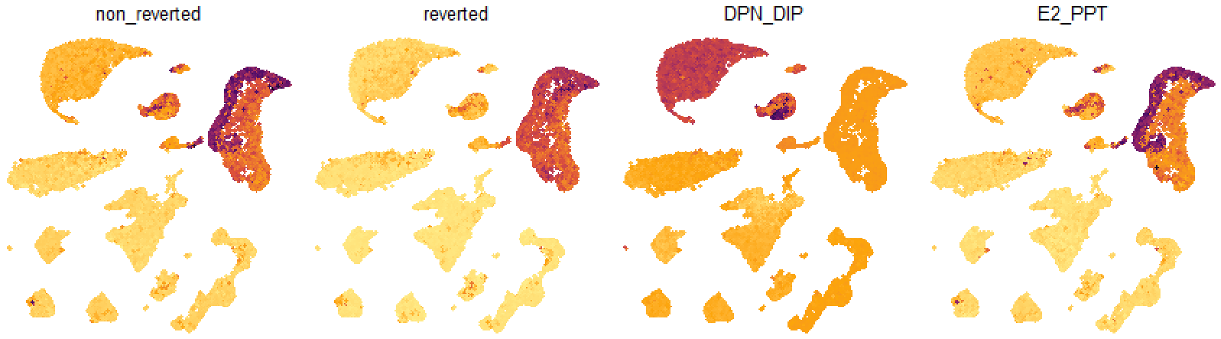
```
# cell type frequency change (%CD - %HFD)
df2 <- df %>%
  dplyr::select(annot, mouseStSt_minus_NAFLD_freq) %>%
  dplyr::filter(!duplicated(.)) %>%
  dplyr::mutate(change=ifelse(mouseStSt_minus_NAFLD_freq>0, 'up', 'down'))
```

```
ggplot(df2, aes(x=mouseStSt_minus_NAFLD_freq, y=annot, fill=change)) +
  geom_bar(color="black", size=0.5, width=0.8, position="stack", stat="identity") +
  scale_fill_manual(values = colPals$UpDown) +
  scale_x_continuous(expand = expansion(mult = c(.0, .1)),
                     limit = c(-25, 25),
                     breaks = c(-20, 0, 20)) +
  xlab('Change (%)') +
  theme_bw() +
  theme(legend.position = 'none')
```



Cell type specificity of ER activation signatures

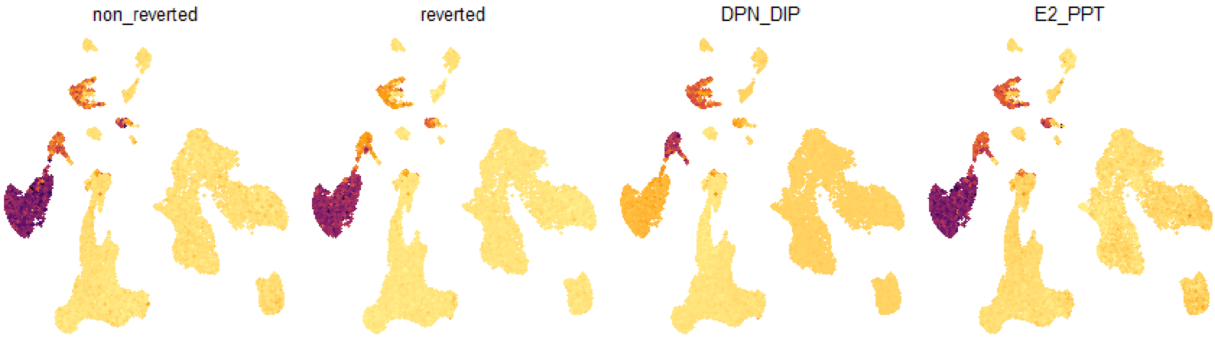
```
# subsample single-cell data to 5000 cells per cell type for enrichment analysis
set.seed(22)
dat <- liver$mouseStSt$All
subs <- dat@meta.data %>%
  tibble::rownames_to_column(var = 'cellid') %>%
  dplyr::left_join(liver_meta, by = 'sample') %>%
  dplyr::filter(sex == 'male' & annot != 'HsPCs') %>%
  dplyr::group_by(annot) %>%
  dplyr::slice_sample(n=5000)
dat <- subset(dat, cells = subs$cellid)

res <- getPAS(x = dat,
              gene.sets = DEG_sets$gene_symbols,
              npcs = 100)

p <- lapply(names(DEG_sets$gene_symbols), function(set) {
  local({
    df <- data.frame(UMAP_1 = res$UMAP_1,
                     UMAP_2 = res$UMAP_2)
    df$score <- res[, set]
    p1 <- ggplot(df, aes(x=UMAP_1, y=UMAP_2, color=score)) +
      geom_point(shape=19, size=1) +
      scale_color_gradientn(colours = colPals$inferno) +
      theme_void() +
      theme(legend.position = 'none',
            plot.title = element_text(hjust = 0.5)) +
      ggtitle(set)
  })
})

patchwork::wrap_plots(p, nrow=1, ncol=4, byrow=T, guides = 'collect')
```



Conservation of cell type specificities in primates

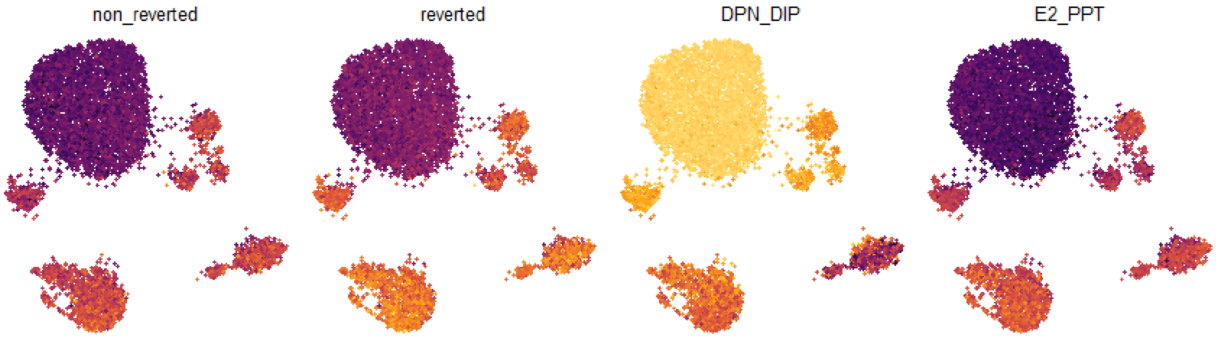
Human

```
# subsample single-cell data to 5000 cells per cell type for enrichment analysis
set.seed(22)
dat <- liver$human$All
subs <- dat@meta.data %>%
  tibble::rownames_to_column(var = 'cellid') %>%
  dplyr::left_join(liver_meta, by = 'sample') %>%
  dplyr::filter(sex == 'male') %>%
  dplyr::group_by(annot) %>%
  dplyr::slice_sample(n=5000)
dat <- subset(dat, cells = subs$cellid)

DEG_sets_human <- lapply(DEG_sets$gene_id, function(x) {
  mouse_human_orthologs %>%
    dplyr::filter(GeneID_mouse %in% x) %>%
    dplyr::pull(GeneSymbol_human) %>%
    unique()
})

res <- getPAS(x = dat,
              gene.sets = DEG_sets_human,
              npcs = 100)

p <- lapply(names(DEG_sets_human), function(set) {
  local({
    df <- data.frame(UMAP_1 = res$UMAP_1,
                     UMAP_2 = res$UMAP_2)
    df$score <- res[, set]
    p1 <- ggplot(df, aes(x=UMAP_1, y=UMAP_2, color=score)) +
      geom_point(shape=19, size=1) +
      scale_color_gradientn(colours = colPals$inferno) +
      theme_void() +
      theme(legend.position = 'none',
            plot.title = element_text(hjust = 0.5)) +
      ggtitle(set)
  })
})

patchwork::wrap_plots(p, nrow=1, ncol=4, byrow=T, guides = 'collect')
```



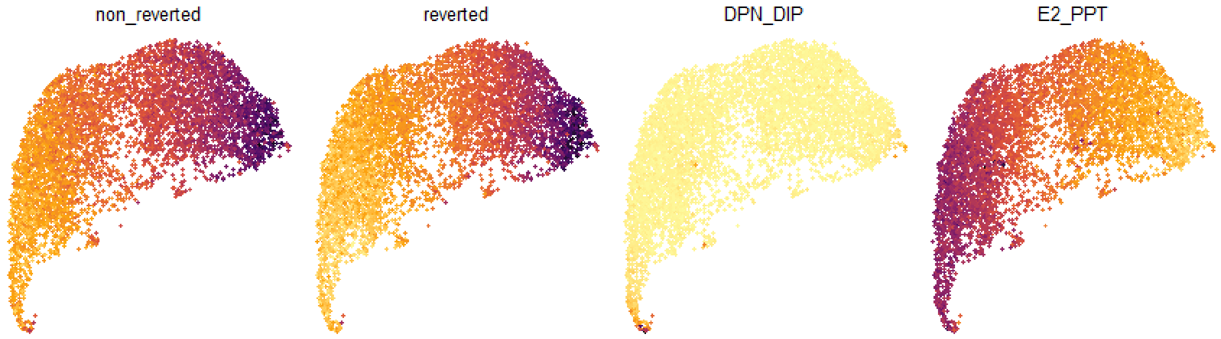
Macaque

```
# subsample single-cell data to 5000 cells per cell type for enrichment analysis
set.seed(22)
dat <- liver$multisp$monkey
subs <- dat@meta.data %>%
  tibble::rownames_to_column(var = 'cellid') %>%
  dplyr::left_join(liver_meta, by = 'sample') %>%
  dplyr::filter(sex == 'male') %>%
  dplyr::group_by(annot) %>%
  dplyr::slice_sample(n=5000)
dat <- subset(dat, cells = subs$cellid)

DEG_sets_macaque <- lapply(DEG_sets$gene_id, function(x) {
  mouse_macaque_orthologs %>%
    dplyr::filter(GeneID_mouse %in% x) %>%
    dplyr::pull(GeneSymbol_macaque) %>%
    unique()
})

res <- getPAS(x = dat,
              gene.sets = DEG_sets_macaque,
              npcs = 100)

p <- lapply(names(DEG_sets_macaque), function(set) {
  local({
    df <- data.frame(UMAP_1 = res$UMAP_1,
                     UMAP_2 = res$UMAP_2)
    df$score <- res[, set]
    p1 <- ggplot(df, aes(x=UMAP_1, y=UMAP_2, color=score)) +
      geom_point(shape=19, size=1) +
      scale_color_gradientn(colours = colPals$inferno) +
      theme_void() +
      theme(legend.position = 'none',
            plot.title = element_text(hjust = 0.5)) +
      ggtitle(set)
  })
})

patchwork::wrap_plots(p, nrow=1, ncol=4, byrow=T, guides = 'collect')
```



Liver zoonation of ER activation signatures (spatial transcriptomics)

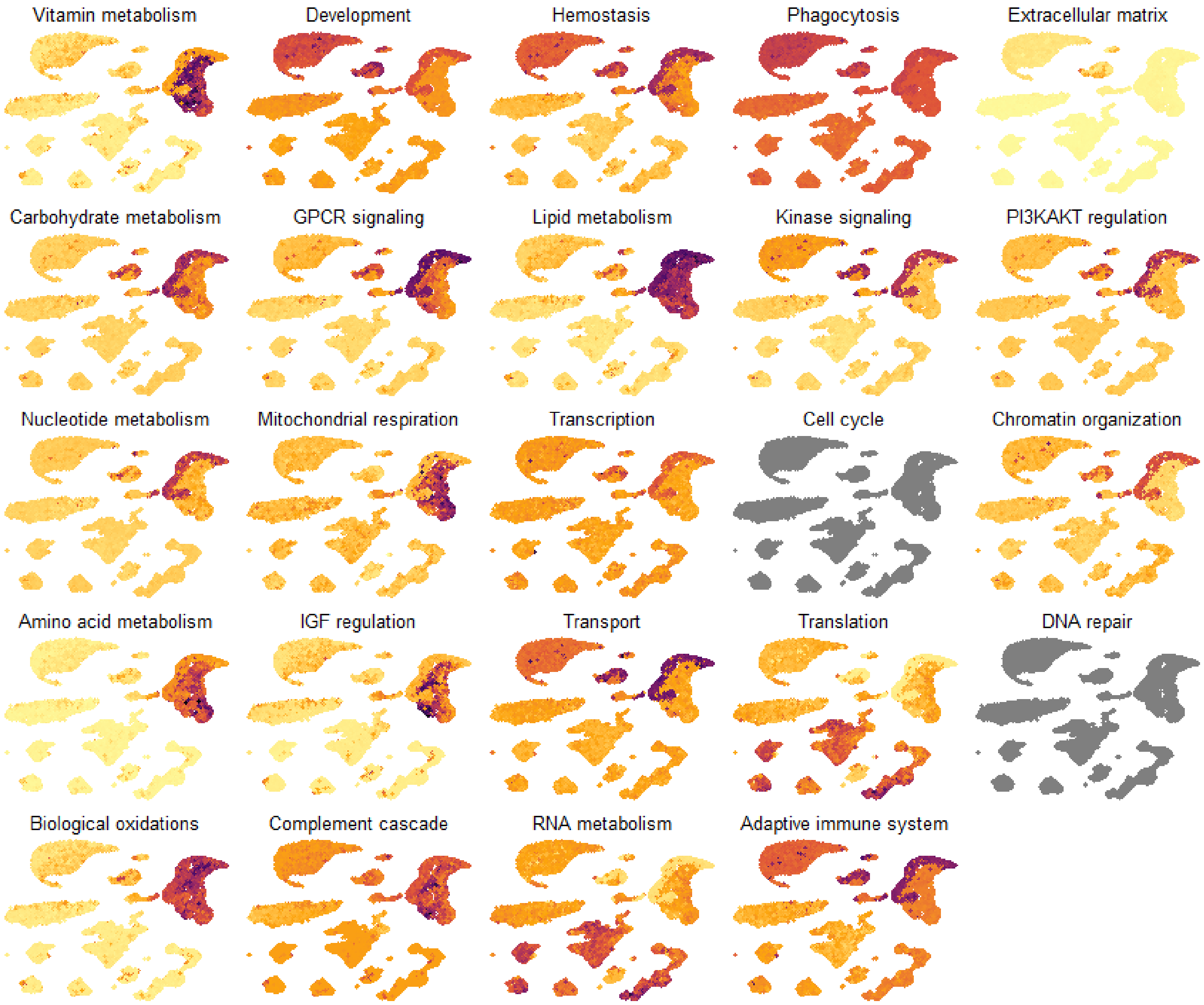
```
# subsample single-cell data to 5000 cells per cell type for enrichment analysis
set.seed(22)
dat <- liver$mouseStSt$Visium
subs <- dat@meta.data %>%
  tibble::rownames_to_column(var = 'cellid') %>%
  dplyr::left_join(liver_meta, by = 'sample') %>%
  dplyr::filter(sex == 'male') %>%
  dplyr::group_by(zonationGroup) %>%
  dplyr::slice_sample(n=5000)
dat <- subset(dat, cells = subs$cellid)

res <- getPAS(x = dat,
              gene.sets = DEG_sets$gene_symbols,
              npcs = 100)

p <- lapply(names(DEG_sets$gene_symbols), function(set) {
  local({
    df <- data.frame(UMAP_1 = res$UMAP_1,
                     UMAP_2 = res$UMAP_2)
    df$score <- res[, set]
    p1 <- ggplot(df, aes(x=UMAP_1, y=UMAP_2, color=score)) +
      geom_point(shape=19, size=1) +
      scale_color_gradientn(colours = colPals$inferno) +
      theme_void() +
      theme(legend.position = 'none',
            plot.title = element_text(hjust = 0.5)) +
      ggtitle(set)
  })
})

patchwork::wrap_plots(p, nrow=1, ncol=4, byrow=T, guides = 'collect')
```



Pathway enrichment across cell types

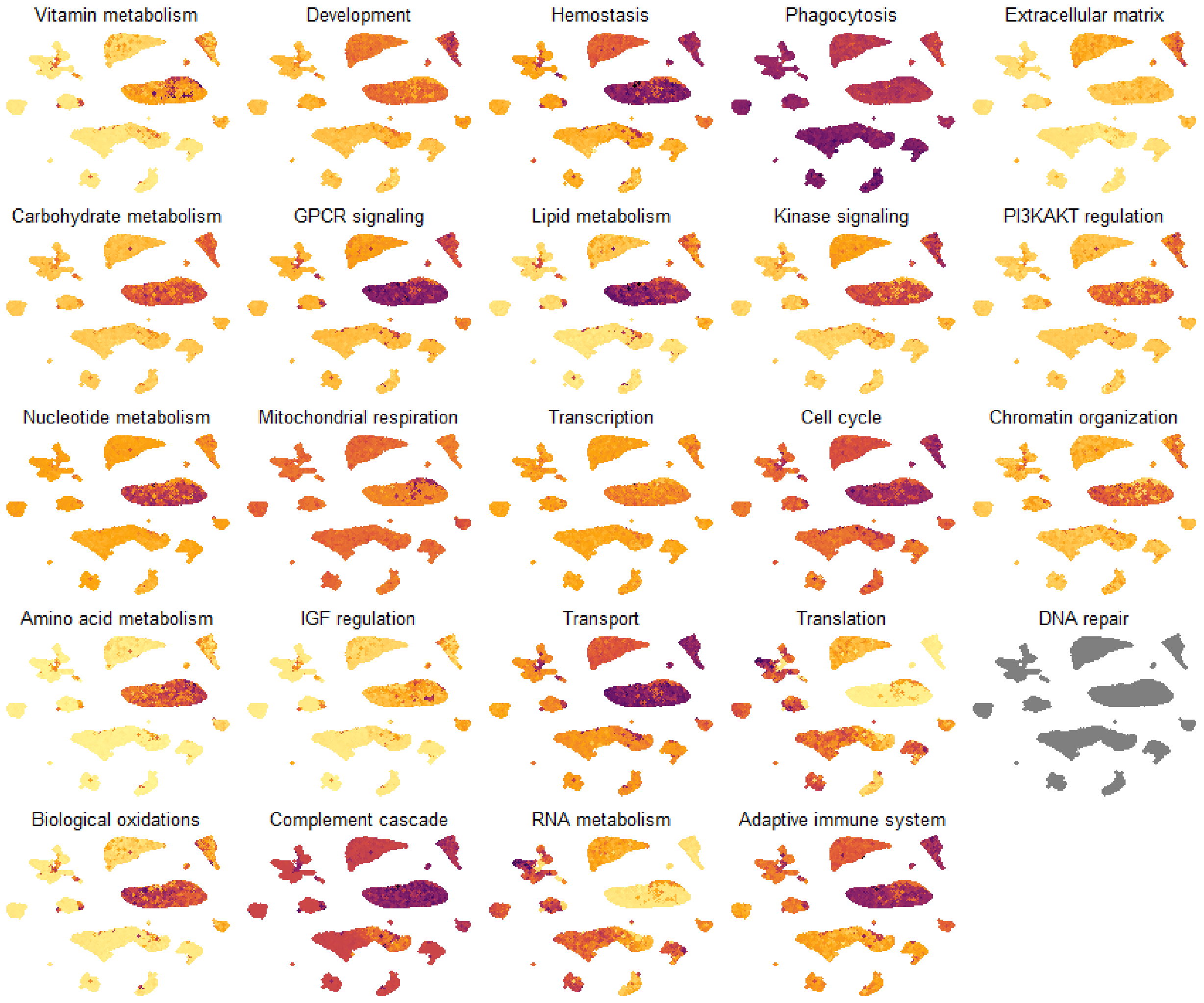
Steady State

```
# subsample single-cell data to 5000 cells per cell type for enrichment analysis
set.seed(22)
dat <- liver$mouseStSt$All
subs <- dat@meta.data %>%
  tibble::rownames_to_column(var = 'cellid') %>%
  dplyr::left_join(liver_meta, by = 'sample') %>%
  dplyr::filter(sex == 'male' & annot != 'HsPCs') %>%
  dplyr::group_by(annot) %>%
  dplyr::slice_sample(n=5000)
dat <- subset(dat, cells = subs$cellid)

res <- getPAS(x = dat,
              gene.sets = pathway_sets,
              npcs = 100)

df <- res %>%
  dplyr::mutate(annot = replace(annot, annot == 'Fibroblasts', 'Stromal cells')) %>%
  dplyr::group_by(annot) %>%
  dplyr::summarize(dplyr::across(`Adaptive immune system`:`Vitamin metabolism`, mean)) %>%
  dplyr::mutate(across(`Adaptive immune system`:`Vitamin metabolism`, ~ (.-min(.)) / (max(.)-min(.)) )) %>%
  tidyr::pivot_longer(cols = `Adaptive immune system`:`Vitamin metabolism`, names_to = 'pathway', values_to = 'score') %>%
  dplyr::mutate(annot = factor(annot, levels = rev(names(colPals$celltypes))),
                pathway = factor(pathway, levels = names(pathway_sets)))

p <- lapply(names(pathway_sets), function(set) {
  local({
    df <- data.frame(UMAP_1 = res$UMAP_1,
                     UMAP_2 = res$UMAP_2)
    df$score <- res[, set]
    p1 <- ggplot(df, aes(x=UMAP_1, y=UMAP_2, color=score)) +
      geom_point(shape=19, size=1) +
      scale_color_gradientn(colours = colPals$inferno) +
      theme_void() +
      theme(legend.position = 'none',
            plot.title = element_text(hjust = 0.5)) +
      ggtitle(set)
  })
})

patchwork::wrap_plots(p, ncol=5, byrow=T, guides = 'collect')
```



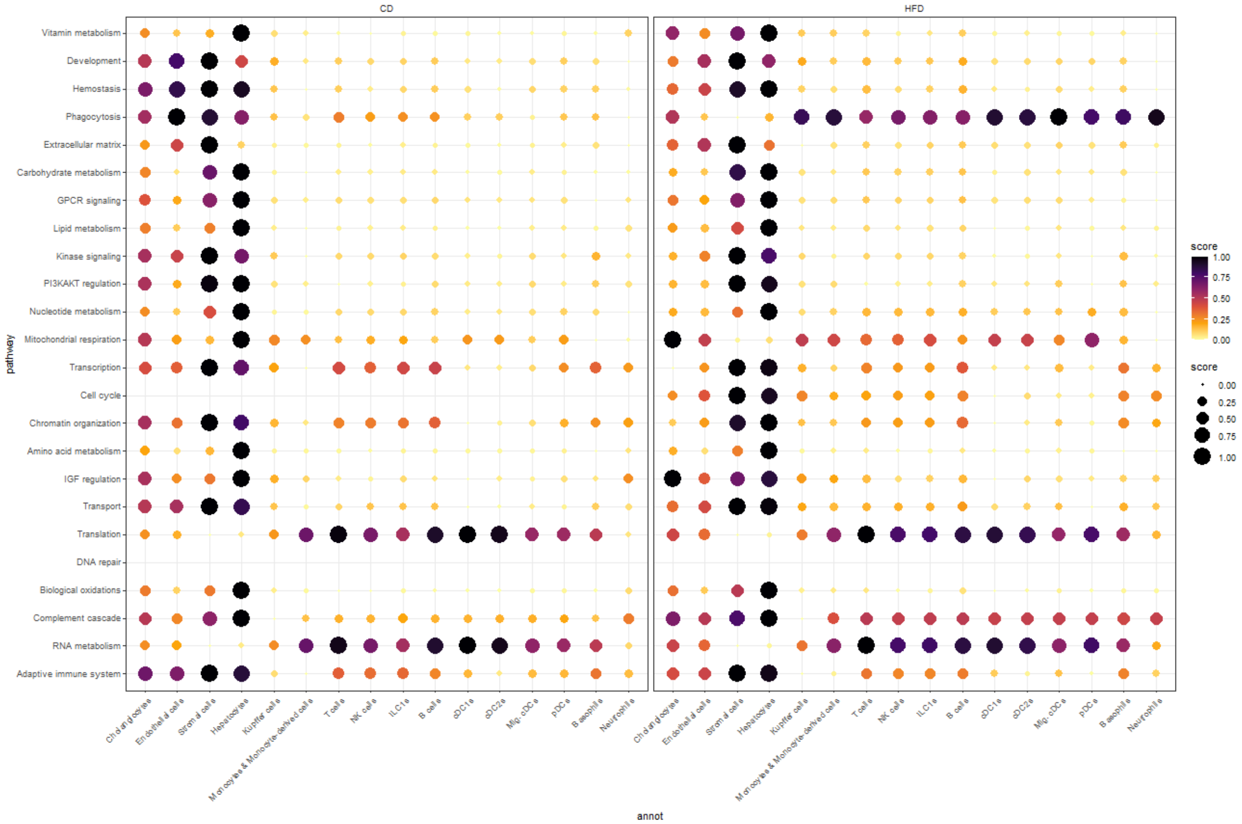
NAFLD

```
# subsample single-cell data to 5000 cells per cell type for enrichment analysis
set.seed(22)
dat <- liver$mouseNAFLD$All
subs <- dat@meta.data %>%
  tibble::rownames_to_column(var = 'cellid') %>%
  dplyr::left_join(liver_meta, by = 'sample') %>%
  dplyr::filter(sex == 'male' & annot != 'HsPCs') %>%
  dplyr::group_by(annot) %>%
  dplyr::slice_sample(n=5000)
dat <- subset(dat, cells = subs$cellid)

res <- getPAS(x = dat,
              gene.sets = pathway_sets,
              npcs = 100)

p <- lapply(names(pathway_sets), function(set) {
  local({
    df <- data.frame(UMAP_1 = res$UMAP_1,
                     UMAP_2 = res$UMAP_2)
    df$score <- res[, set]
    p1 <- ggplot(df, aes(x=UMAP_1, y=UMAP_2, color=score)) +
      geom_point(shape=19, size=1) +
      scale_color_gradientn(colours = colPals$inferno) +
      theme_void() +
      theme(legend.position = 'none',
            plot.title = element_text(hjust = 0.5)) +
      ggtitle(set)
  })
})

patchwork::wrap_plots(p, ncol=5, byrow=T, guides = 'collect')
```

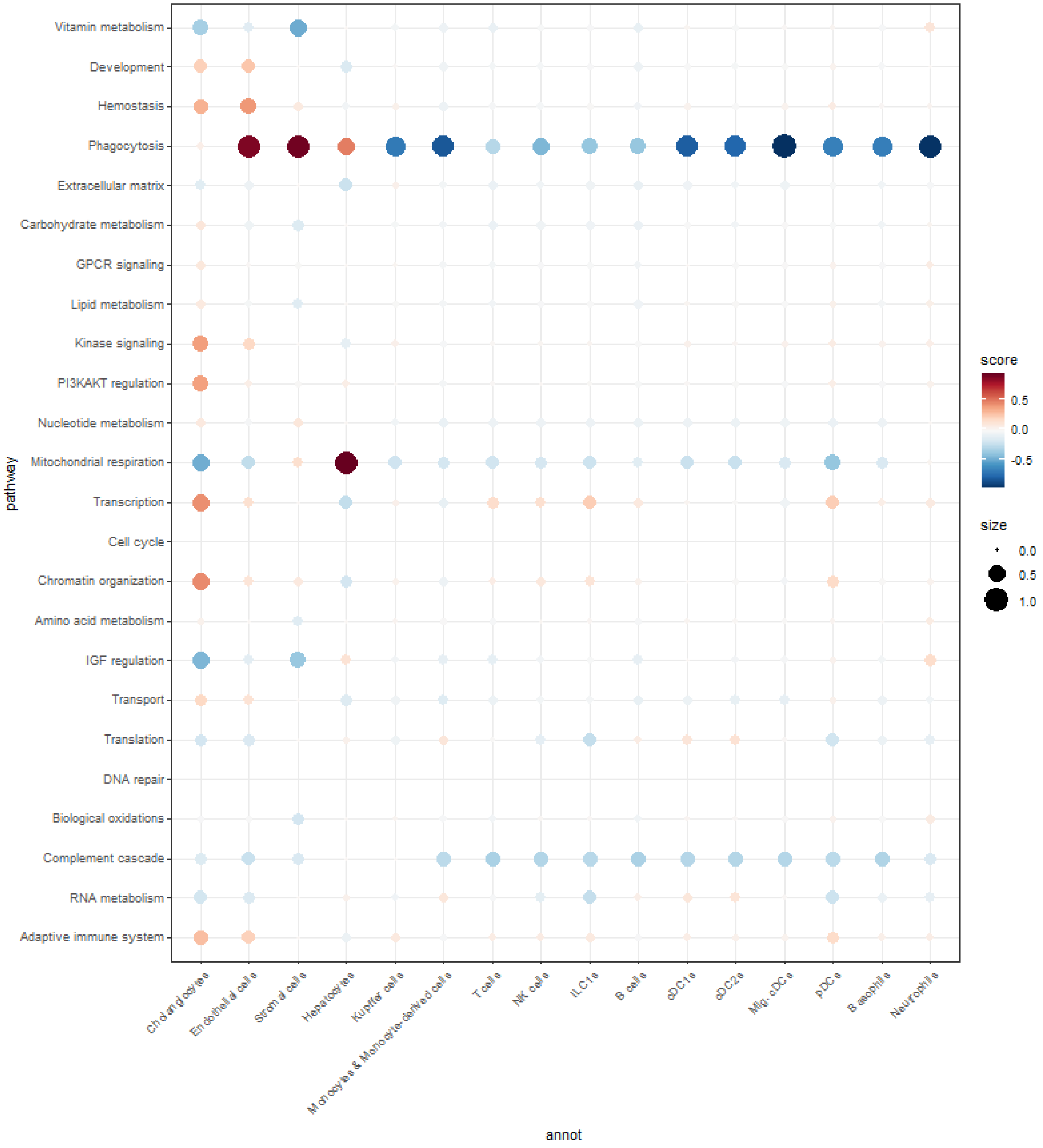


Comparison of steady state vs NAFLD

```
df2 <- res %>%
  dplyr::mutate(annot = replace(annot, annot == 'KCs', 'Kupffer cells')) %>%
  dplyr::mutate(annot = replace(annot, annot == 'Mono/mono-derived cells', 'Monocytes & Monocyte-derived cells')) %>%
  dplyr::mutate(annot = replace(annot, annot == 'Mig cDCs', 'Mig. cDCs')) %>%
  dplyr::group_by(annot) %>%
  dplyr::summarize(dplyr::across(`Adaptive immune system`:`Vitamin metabolism`, mean)) %>%
  dplyr::mutate(dplyr::across(`Adaptive immune system`:`Vitamin metabolism`, ~ (.-min(.)) / (max(.)-min(.)) )) %>%
  tidyr::pivot_longer(cols = `Adaptive immune system`:`Vitamin metabolism`, names_to = 'pathway', values_to = 'score') %>%
  dplyr::mutate(cond = 'HFD')

df3 <- df %>%
  dplyr::mutate(cond = 'CD') %>%
  dplyr::bind_rows(df2) %>%
  dplyr::mutate(cond = factor(cond, levels = c('CD','HFD')),
                annot = factor(annot, levels = names(colPals$celltypes)),
                pathway = factor(pathway, levels = rev(names(pathway_sets))))

ggplot(df3, aes(x=annot, y=pathway, size=score, color=score)) +
  geom_point() +
  scale_color_gradientn(colours = colPals$inferno) +
  scale_size(range = c(1,8))+
  facet_wrap(~cond) +
  theme_bw() +
  theme(strip.background = element_blank(),
        axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1),
        axis.text.y = element_text(hjust = 1, vjust = 0.3))
```

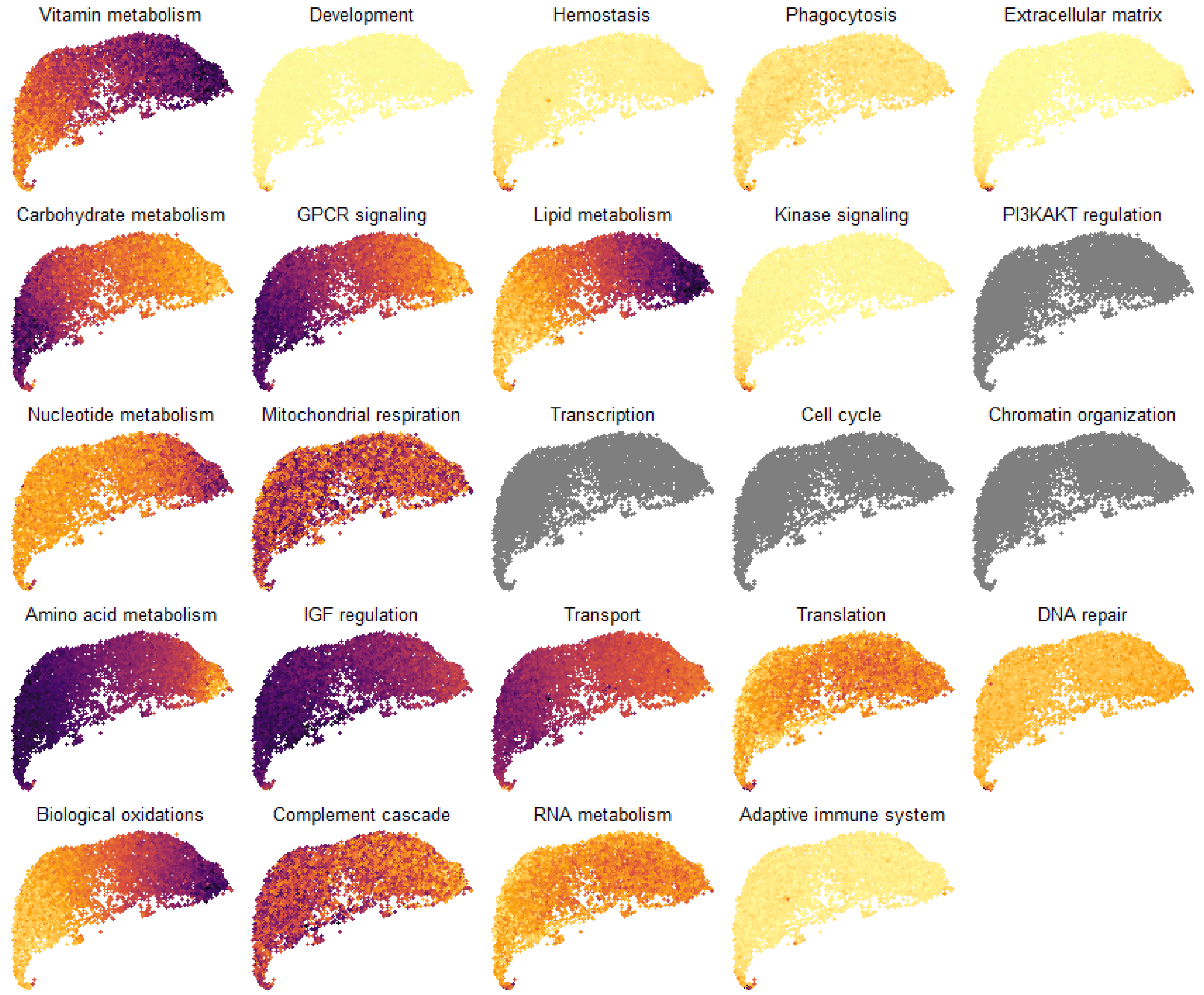


```
# Cell type enrichment change (CD - HFD)
df4 <- df %>%
  dplyr::mutate(score = score - df2$score) %>%
  dplyr::mutate(size = abs(score)) %>%
  dplyr::mutate(annot = factor(annot, levels = names(colPals$celltypes)),
                pathway = factor(pathway, levels = rev(names(pathway_sets))))

ggplot(df4, aes(x=annot, y=pathway, size=size, color=score)) +
  geom_point() +
  scale_color_gradientn(colours = colPals$RdBu) +
  scale_size(range = c(1,8), breaks = c(0,0.5,1), limits = c(0,1))+
  theme_bw() +
  theme(strip.background = element_blank(),
        axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1),
        axis.text.y = element_text(hjust = 1, vjust = 0.3))
```



Pathway zonation in the liver

```
# subsample single-cell data to 5000 cells per cell type for enrichment analysis
set.seed(22)
dat <- liver$mouseStSt$Visium
subs <- dat@meta.data %>%
  tibble::rownames_to_column(var = 'cellid') %>%
  dplyr::left_join(liver_meta, by = 'sample') %>%
  dplyr::filter(sex == 'male') %>%
  dplyr::group_by(zonationGroup) %>%
  dplyr::slice_sample(n=5000)
dat <- subset(dat, cells = subs$cellid)

res <- getPAS(x = dat,
              gene.sets = pathway_sets,
              npcs = 100)

p <- lapply(names(pathway_sets), function(set) {
  local({
    df <- data.frame(UMAP_1 = res$UMAP_1,
                     UMAP_2 = res$UMAP_2)
    df$score <- res[, set]
    p1 <- ggplot(df, aes(x=UMAP_1, y=UMAP_2, color=score)) +
      geom_point(shape=19, size=1) +
      scale_color_gradientn(colours = colPals$inferno) +
      theme_void() +
      theme(legend.position = 'none',
            plot.title = element_text(hjust = 0.5)) +
      ggtitle(set)
  })
})

patchwork::wrap_plots(p, ncol=5, byrow=T, guides = 'collect')
```



SessionInfo

```r
sessionInfo()
```

```
## R version 4.0.5 (2021-03-31)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
## 
## Matrix products: default
## 
## locale:
## [1] LC_COLLATE=English_United States.1252 
## [2] LC_CTYPE=English_United States.1252   
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C                          
## [5] LC_TIME=English_United States.1252    
## 
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base     
## 
## other attached packages:
##  [1] patchwork_1.1.1    pagoda2_1.0.2      igraph_1.2.6       Matrix_1.4-1      
##  [5] SeuratObject_4.0.4 Seurat_4.0.2       forcats_0.5.1      stringr_1.4.0     
##  [9] dplyr_1.1.2        purrr_0.3.4        readr_2.1.2        tidyr_1.2.0       
## [13] tibble_3.2.1       ggplot2_3.3.3      tidyverse_1.3.1   
## 
## loaded via a namespace (and not attached):
##   [1] utf8_1.1.4               reticulate_1.18          R.utils_2.10.1          
##   [4] tidyselect_1.2.0         htmlwidgets_1.5.3        grid_4.0.5              
##   [7] BiocParallel_1.22.0      Rtsne_0.16               munsell_0.5.0           
##  [10] codetools_0.2-18         ica_1.0-3                future_1.21.0           
##  [13] miniUI_0.1.1.1           withr_2.5.0              spatstat.random_2.2-0   
##  [16] colorspace_2.0-0         Biobase_2.48.0           highr_0.10              
##  [19] knitr_1.31               rstudioapi_0.13          stats4_4.0.5            
##  [22] ROCR_1.0-11              tensor_1.5               distillery_1.2-1        
##  [25] listenv_0.9.0            labeling_0.4.2           urltools_1.7.3          
##  [28] polyclip_1.10-0          farver_2.0.3             extRemes_2.1-3          
##  [31] parallelly_1.36.0        vctrs_0.6.3              generics_0.1.3          
##  [34] xfun_0.31                fastcluster_1.2.3        R6_2.5.1                
##  [37] locfit_1.5-9.4           flexmix_2.3-19           spatstat.utils_2.3-0    
##  [40] RcppArmadillo_0.11.0.0.0 assertthat_0.2.1         promises_1.1.1          
##  [43] scales_1.2.1             nnet_7.3-15              gtable_0.3.3            
##  [46] Cairo_1.5-15             globals_0.16.2           goftest_1.2-3           
##  [49] drat_0.2.3               rlang_1.1.1              MatrixModels_0.4-1      
##  [52] splines_4.0.5            lazyeval_0.2.2           spatstat.geom_2.4-0     
##  [55] broom_0.8.0              brew_1.0-8               yaml_2.2.1              
##  [58] reshape2_1.4.4           abind_1.4-5              modelr_0.1.8            
##  [61] backports_1.4.1          httpuv_1.5.5             tools_4.0.5             
##  [64] sccore_1.0.1             ellipsis_0.3.2           spatstat.core_2.4-2     
##  [67] RColorBrewer_1.1-3       BiocGenerics_0.36.1      ggridges_0.5.3          
##  [70] Rcpp_1.0.7               plyr_1.8.6               dendsort_0.3.4          
##  [73] rpart_4.1-15             deldir_1.0-6             pbapply_1.7-2           
##  [76] cowplot_1.1.1            scde_2.18.0              zoo_1.8-10              
##  [79] haven_2.5.0              ggrepel_0.9.1            cluster_2.1.1           
##  [82] fs_1.5.2                 magrittr_2.0.3           data.table_1.13.6       
##  [85] scattermore_0.8          SparseM_1.81             p2data_1.0.0            
##  [88] lmtest_0.9-40            triebeard_0.3.0          reprex_2.0.1            
##  [91] RANN_2.6.1               pcaMethods_1.82.0        fitdistrplus_1.1-11     
##  [94] matrixStats_0.58.0       hms_1.0.0                mime_0.9                
##  [97] evaluate_0.21            xtable_1.8-4             RMTstat_0.3.1           
## [100] readxl_1.4.0             N2R_1.0.1                gridExtra_2.3           
## [103] compiler_4.0.5           KernSmooth_2.23-18       crayon_1.5.1            
## [106] R.oo_1.24.0              htmltools_0.5.2          mgcv_1.8-40             
## [109] later_1.1.0.1            tzdb_0.3.0               lubridate_1.8.0         
## [112] DBI_1.1.3                dbplyr_2.1.1             MASS_7.3-53             
## [115] cli_3.6.1                R.methodsS3_1.8.2        parallel_4.0.5          
## [118] pkgconfig_2.0.3          plotly_4.9.3             spatstat.sparse_2.1-1   
## [121] xml2_1.3.3               rvest_1.0.2              digest_0.6.27           
## [124] sctransform_0.3.3        RcppAnnoy_0.0.19         spatstat.data_2.2-0     
## [127] rmarkdown_2.14           cellranger_1.1.0         leiden_0.3.7            
## [130] Rook_1.1-1               uwot_0.1.11              edgeR_3.32.1            
## [133] modeltools_0.2-23        shiny_1.6.0              Lmoments_1.3-1          
## [136] quantreg_5.88            rjson_0.2.20             lifecycle_1.0.3         
## [139] nlme_3.1-151             jsonlite_1.8.0           viridisLite_0.4.2       
## [142] limma_3.44.3             fansi_0.4.2              pillar_1.9.0            
## [145] lattice_0.20-41          fastmap_1.1.0            httr_1.4.2              
## [148] survival_3.2-7           glue_1.4.2               png_0.1-7               
## [151] stringi_1.5.3            irlba_2.3.5              future.apply_1.7.0
```